

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently amended) A method for ordered command verification testing of a multiprocessor computer system comprising multiple parallel processor threads, comprising:
 - | providing at least one rule for forming a legal ordered ~~sequencees~~ sequence of semaphore test commands;
 - | forming a plurality of buckets of semaphore commands each comprising a plurality of the commands arranged in different sequentially ordered command sequences legal under the at least one rule;
 - | randomly selecting, ordering and combining a plurality of the buckets in a sequential bucket test combination having a test sequential order, the sequential bucket test combination having an ordered composite test command sequence, wherein an order of any of the buckets relative to another of the buckets within the test sequential order may be changed within the test sequential order, the test sequential order legal under the at least one rule and thereby generating a predictable result when executed by the threads;
 - | a semaphore manager sequentially ~~distribution~~ distributing at least one of the composite test sequence commands to a first plurality of the processor threads in a first thread distribution; and
 - | observing the performance of each of the first plurality of threads in response to the distributed test commands.

2. (Previously presented) The method of claim 1, wherein the executable semaphore test commands may be selected from the group comprising ordered arguments of the computer system real operational code, ordered specific test instructions targeted for verification purposes, and randomly generated ordered instructions.

3. (Previously presented) The method of claim 2, further comprising:
including a wait command in the composite test command sequence test commands;
the wait command causing the semaphore manager to pause for at least one instruction cycle and skip a next first plurality thread and thereby distribute at least one next command ordered subsequent to the wait command to an alternative thread, the manager thereby sequentially distributing the test combination commands to each of a second plurality of the processor threads different from the first plurality; and
observing the performance of each of the second plurality of threads in response to the distributed test commands.

4. (Previously presented) The method of claim 3 wherein combining the buckets further comprises including said wait command within a bucket, between first and second buckets, before the first bucket, or after the second bucket.

5. (Previously presented) The method of claim 3 further comprising the step of randomly inserting the wait command within the composite test command sequence.

6. (Cancelled)

7. (Previously presented) The method of claim 5 wherein including said wait command comprises inserting a random amount of the wait command.

8. (Cancelled)

9. (Previously presented) The method of claim 7, further comprising:
providing a parameter for at least one of the plurality of executable commands;
assigning a value to the parameter; and
identifying and selecting a next thread for a next at least one command in response to the parameter value.

10. (Cancelled)

11. (Previously presented) The method of claim 9 further comprising randomly selecting and unlocking a semaphore in response to the parameter value.

12-25 (Cancelled)